TRENDY EKONOMIKY A MANAGEMENTU
TRENDS ECONOMICS AND MANAGEMENT
ISSN 1802-8527 (Print) / ISSN 2336-6508 (Online)

2017 28(1): 9–21
DOI: http://dx.doi.org/10.13164/trends.2017.28.9

Hyperparameter Optimization of Artificial Neural Network in Customer Churn Prediction using Genetic Algorithm

Martin Fridrich

Abstract

Purpose of the article: The ability of the company to predict customer churn and retain customers is considered to be worthy competitive advantage since it improves cost allocation in customer retention programs, retaining future revenue and profits. In addition, it has several positive indirect impacts such as increasing customer's loyalty. Therefore, the focus of the article is on building highly reliable and robust classification model, which deals with such a task.
Methodology/methods: The analysis is carried out on labelled ecommerce retail dataset describing 10 000 most valuable customers with the highest CLV (Customer Lifetime Value). To obtain the best performing ANN (Artificial Neural Network) classification model, proposed hyperparameter search space is explored with genetic algorithm to find suitable parameter settings. ANN classification performance is measured with regard to prediction ability, which is understood as point estimate of AUC (Area Under Curve) mean on 4fold cross-validation set. Explored part of hyperparameter search space is analyzed with conditional inference tree structure addressing underlying fundamental context of given optimization which results in identification of critical factors leading to well performing ANN classification model.
Scientific aim: To present and execute experimental design for performance evaluation and hyperparameter optimization of classification models, which are used for customer churn prediction.
Findings: It is concluded and statistically proven that in experimental context described, regularization parameter as well as training function have significant influence on classifiers AUC performance contrasting other properties of ANN. More specifically, well performing ANN classification models have regularization parameter set to 0, adaptation function set to trainlm or trainscg and more than 100 training epochs. Global optimum is identified for solution with regularization parameter set to 0, trainlm adaptation function, 350 training epochs and 7-4-2 architecture.
Conclusions: Results imply that placing hyperparameter optimization to ANN classification model leads to improved customer churn prediction ability. The article describes design and execution of machine learning pipeline, hyperparameter optimization and original meta-analysis of the results with conditional inference tree structure, which are considered beneficial for further research.

Keywords: customer churn, machine learning, artificial neural networks, genetic algorithm, hyperparameter optimization

JEL Classification: M1, M3, C38

Introduction

In highly competitive environment it is common that achieving business organizations goals relies on its ability of customer relationship management or relationship management with other interested parties. Colgate, Danaher (2000) and Poel, Lariviere (2004) show that in saturated markets, customer acquisition is very expensive and demanding process. Therefore, customer retention programs, as part of customer relationship management, are becoming topic of interest among companies. Customer retention management in general consists of activities related to customer churn prediction, customer retention (benefit) program, retention uplift modeling and other activities linked to customer defection. To allocate retention activities and costs efficiently it is crucial for customer defection prediction to be as reliable and precise as possible.

Popular techniques used for customer churn prediction are logistic regression, decision tree, fuzzy logic, Bayesian classifier, SVM, and neural networks (Ngai *et al.*, 2009; Kumar, Garg, 2013), the paper aims at application of the last one. Artificial neural networks are chosen due to proven prediction ability in the domain of customer churn when compared to other base techniques (Hadden *et al.*, 2006; Iwata *et al.*, 2006; Ngai *et al.*, 2009; Kumar, Garg, 2013). Although prediction performance as key metric is considered with application of artificial neural networks, other aspects of prediction model are suppressed – such as interpretability, which is important for fundamental understanding of customer churn.

1. Preliminaries

Preliminaries section consists of short description of building blocks used in machine learning pipeline leading to artificial neural network classifier. Principle component analysis is used to reduce number of exploratory variables in customer churn prediction task and to consequentially speed up learning process of artificial neural network. Classification itself is dealt with by artificial neural network. Experimental parameter tuning is considered as optimization task, therefore search through parameter space is carried out with genetic algorithm. Relationship between architecture and training parameters of neural network and its performance is further analyzed with conditional inference tree.

1.1 Principle component analysis

Principle component analysis (PCA) is unsupervised technique for dimensionality reduction, more precisely for projecting data into its lower-dimensional representation while capturing most of its variation. The idea is that each of the n observations lives in p-dimensional space, but not all for these dimensions capture same amount of variability. PCA searchers for small number of dimensions that capture the most of variability, returned dimensions are linear combination of p features (Rogers, Girolami, 2012; Garet *et al.*, 2013).

Reducing dimensionality is common goal for feature extraction, reducing autocorrelation and visualization of complex data. Main motivation is to reduce computational cost, however PCA might negatively impacts classifiers performance (Jain *et al.*, 2000).

1.2 Genetic algorithm

Genetic algorithm (GA) is metaheuristic optimization method, which belongs to broad class of evolutionary algorithms. Natural selection inspires GA to mimic evolution (survival of the fittest). It is based on iterative process starting from initial set of solutions (population), where every solution is given set of properties (genes). Initialization is followed by evaluation of optimization objective (fitness function) across the population, where the fittest solutions are selected (selection) and their properties recombined resulting in new population (crossover). To

overcome convergence to local minima, chance of random property shift (mutation) between generations is introduced. Process of selection, crossover and mutation is sequentially repeated until computational limits are exceeded or optimization objective of fitness function is achieved (Mitchell, 1996; Dostál, 2012).

GA are used to solve mostly global optimization objectives over problem domains with complex parameter space, where are successful in overcoming local optimum compared to gradient based algorithms. Real world applications include and are not limited to traveling salesman, assembly line optimization, antenna design. Critics object that GA is adding more complexity to original problem and in non-trivial applications can be inefficient (Mitchell, 1996; Skiena, 1998).

1.3 Artificial neural network

Artificial neural network (ANN) is supervised learning technique. It can be described as simplified mathematical model inspired by biological neural network of living species. It is based on processing information by many simple elements (neurons), passing signal between neurons over connection (link) with associated weight, each neuron has its activation function (nonlinear) for processing inputs into output signal. ANN is described with its architecture (number of layers, neurons), training/learning method (how connection weight is calculated) and activation function (Fausett, 1994).

ANNs are broadly used to solve classification problems, pattern recognition, clustering, or constrained optimization problem. Their increasing popularity is related to proven ability of generalizing complex functions, robustness considering data-preprocessing and rise of accessible computational power. However underlying mechanism of trained network is hard to interpret in fundamental context of the problem therefore is usually considered as black box (Fausett, 1994; Dostál, 2012).

1.4 Conditional inference tree

Conditional inference tree belongs into non-parametric class of regression/classification trees combining conditional inference applications with tree-structured regression/classification models. Regression tree is built with separation of data into subsets (leaves), by finding splits (nodes), which separates data in the best possible way, approach is applied on subsets until getting to pure subsets. Tree-regression models tend to suffer from two issues – overfitting (can be dealt with pruning) and bias towards selected variables affecting interpretability of tree-structured model. As response to these problems – conditional inference trees were proposed by Hothorn *et al.* (2006), where stopping criterion is based on resampling and multiple inference tests.

Decision trees are easy to interpret and can deal with non-linear classification and regression problem domains. Prediction performance of base conditional inference trees has been proven on similar level as with pruned regression trees with no bias towards selected variables (Horton *et al.*, 2006; Horton *et al.*, 2015).

2. Experiment design and implementation

The paper aim is to develop prediction model, which can identify customers at risk of defection with regard to improvement of chosen performance metrics. In this article, problem is understood as binary classification and is dealt with by ANN classifier. Hyperparameter optimization of ANN is treated as mixed integer optimization problem and is carried out by genetic algorithm. Moreover, obtained results are analyzed to determine which setting leads to the best prediction. Bearing in mind reproducibility of results, consistence and usability in production settings, machine learning pipeline is proposed in Figure 1, where elements are discussed later in more detail. Whole pipeline is

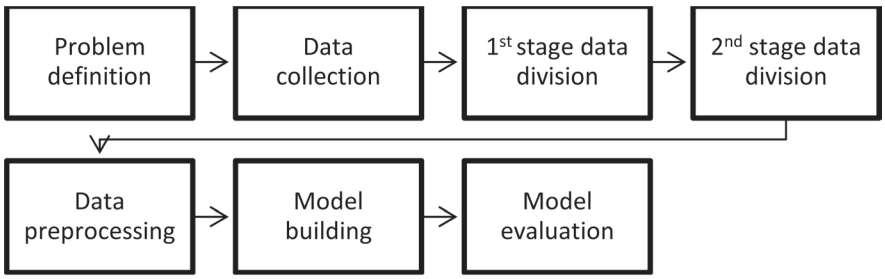

Figure 1. Machine learning pipeline applied. Source: Author's own work.

implemented in MathWorks Matlab 2016a using Neural Networks Toolbox 9.0, Global Optimization Toolbox 7.4 and Parallel Computing Toolbox 6.8. Further analysis and visualization of experimental results are done in R 3.3.0, R Studio 1.0.44 and partykit 1.0.5 package.

2.1 Data collection

Dataset was obtained from analytical database of e-commerce company Alza.cz. It contains customer interactions with the company till 03/2014, included. Customer defection is understood as no new purchase for 9 consecutive months (03/2014–09/2014), hence only complete degree of defection is considered.

For modelling purposes 10 000 customers with the highest Customer Lifetime Value (CLV) is chosen, also with class balance between churned and non-churned customers in mind (50 % of defected customers). Expected cumulative CLV for next 3 years with no changes in churn rate was estimated to 60 M CZK. This estimation illustrates both potential in customer relationship management and financial gap for retention management.

Only customer interactions with company are used as primary features, based on cited research and broadly applied RFM segmentation. All base features are aggregated with period of 3, 6, 12 months and whole customer lifecycle.

2.2 Data division

Data division is important part in experiment design responsible for proper performance evaluation and addressing bias-variance tradeoff. In the paper two stages of data division are incorporated. Stratification to balance both churned and non-churned customer is used in both stages.

In first stage, whole dataset is divided into two parts - cross validation set (90 % of data) and testing set (10 % of data). Cross validation set is passed to second stage of machine learning pipeline, testing set is used to evaluate real world performance on unseen data.

In second stage, cross validation set is used as input into 4-fold cross validation division. This technique splits cross validation set into 4 equally sized folds, where each

Table 1. Primary explanatory variables.

Variable	Description	DB data type	Unit
RevAvg	Average revenue on invoice	Money	[CZK]
InvCount	Number of invoices issued	Int	[n]
RecCount	Number of claims made	Int	[n]
ComCount	Number of contacts with customer service	Int	[n]

Source: Author's own work.

combination of 3 folds are used as training set (without repetition) and each 1 set used as validation set.

2.3 Data preprocessing

Prior to PCA training features are normalized with z-score, as it allows PCA to compare primary features on scale of same magnitude. Z-score calculation on training set also results in population μ and σ parameter estimations which are used for further scaling of validation and test sets.

PCA threshold for captured variability is set to at least 95 %. PCA coefficients are derived from training set and are used to further extraction from validation and test set. Strict split among training, validation and test sets and sequence of processing is crucial for prevention of data leakage and model building and evaluation.

2.4 Model building

2.4.1 Genetic algorithm

Chromosome structure embodies properties of ANN classifier and is represented in double vector scheme, where each parameter is encoded as integer value pointer to ANN parameter vector. Other solution might be to encode chromosome in bit string scheme, however in mixed integer problem domain, prior solution is considered to be more advisable (Herrera *et al.*, 1998).

To generate new population and introduce changes to chromosomes Laplace crossover operator and Power mutation operator are used. To ensure integer constrains hold after crossover and mutation steps, truncation procedure is carried out (Deep *et al.*, 2009). Binary tournament selection is chosen as it has better convergence and computational time complexity when compared to other reproduction operators (Goldberg, Deb, 1991). With regard to both size of searched parameter space and computational complexity of task at hand, population size is set to 16 and number of generations to 32, resulting in training and evaluation of 512 ANN models.

Objective function is constructed as penalty function with respect to feasibility of solution. (i) If two feasible solutions are compared, the one minimizing fitness function

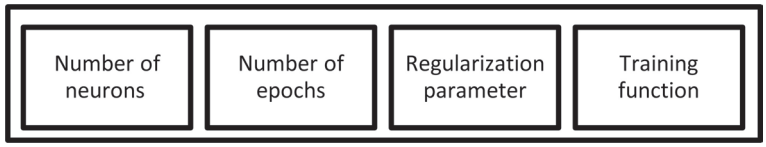

Figure 2. Chromosome coding scheme. Source: Author's own work.

Table 2. GA parameter specifications.

Parameter	Value
Population type	Double vector
Objective scaling	Rank
Elite children	2
Termination conditions	32 generations
Population size	16
Selection operator	Binary tournament
Crossover operator	Laplace crossover
Crossover fraction	0.9
Mutation operator	Power mutation

Source: Author's own work.

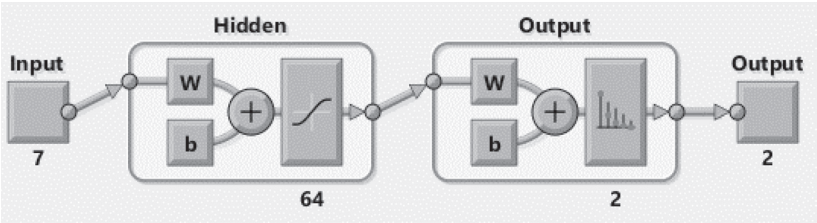

Figure 3. 7-64–2 architecture with Tan-Sig and Softmax activation functions, Matlab 2016a. Source: Author's own work.

Table 3. ANN parameter vectors.

Parameter	Vector							
Number of neurons	2	4	8	16	32	64	128	256
Number of epochs	50	100	150	200	250	300	350	400
Regularization parameter	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35
Training function	trainlm	trainscg	traincgb	trainrp				

Source: Author's own work.

is chosen. Penalty function value is set to its fitness function value. (ii) If feasible and infeasible solutions are compared, feasible solution is chosen. Penalty function value is set to its fitness function value. (iii) If two infeasible solutions are compared, solution with lower constraint violation is chosen. Penalty function value is set to maximum population fitness function value (Deb, 2009). Internal fitness function is based on 1-AUC, where AUC is evaluated on cross-validation set. Please see details of AUC metric under chapter 2.5. Population objective values are ranked through each generation to scale spread of its values. Internal fitness function is implemented through vectorized approach enabling evaluation of whole generation in parallel.

2.4.2 Artificial neural network

Feedforward ANN with two layers and backpropagation is used, with Tan-Sigmoid activation function in hidden layer and Softmax activation function in output layer. Tan-sig function is used as it performs better with zero-centered mean inputs when compared to other non-linear activation functions (Montavon, 2012). Softmax function naturally transforms output from hidden layer to categorical probability distribution in output layer, hence is suitable for classification problems (Bishop, 2006).

Other properties such as number of neurons in hidden layer, training epochs, specific training function and regularization parameter are subject to hyperparameter tuning. Training cost function is set to MSE due its wide compatibility among training functions. Initial weights of ANN are fixed with pseudo-random number generator. Permutation vector for each parameter applied is shown on Table 2, resulting in 2048 models for whole parameter space.

2.5 Model evaluation

Classifier performance assessment is obtained with confusion matrix; example is shown in Table 3. In this case, null hypothesis is defined as customer retains, alternative hypothesis is then considered as customer churns. Common measures derived from confusion matrix are Accuracy, Precision, Recall and F score. Paper aims at multiple model's comparison, therefore only Accuracy and Area Under Curve (AUC) are applied.

Accuracy is used for clear interpretability, although it is not reliable while dealing with class

Table 4. Confusion matrix concept.

Prediction/Reality	Churned	Non-churned
Churned	True Positives	False Positives
Non-churned	False Negatives	True Negatives

Source: Fawcett, 2006.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives} \quad (1)$$

$$True\ Positive\ Rate = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (2)$$

$$False\ Positive\ Rate = \frac{False\ Positives}{True\ Negatives + False\ Negatives} \quad (3)$$

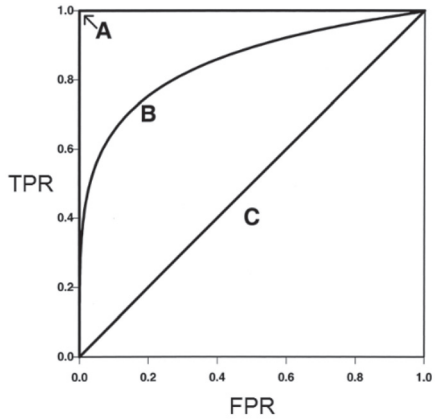

Figure 4. Three hypothetical ROC curves representing the diagnostic accuracy of the perfect classification (lines A; AUC=1) on the upper and left axes in the unit square, a typical ROC curve (curve B; AUC=0.85), and a diagonal line corresponding to random chance (line C; AUC=0.5). Source: Zoe et al., 2007.

imbalance – (1). To overcome that AUC is suggested as ultimate metric for performance evaluation. AUC is based on Receiver Operator Characteristics (ROC) curve and benefits from its evaluation of tradeoff between True Positive Rate (TPR) – (2) and False Positive Rate (FPR) – (3) while shifting criterion threshold (Fawcett, 2006; Zou *et al.*, 2007).

3. Experimental results

ML pipeline implementation with parallelized optimization was carried out on Intel Core i7-6700K CPU, 16GB RAM, Windows 10 64-bit machine and took 24h 25min to process, which is improvement when compared to ML pipeline with parallelized grid search on same machine resulting in 75h 34min.

Optimization progress illustrated by fitness function is addressed with Figure 5. Spikes in mean fitness value can be interpreted as widening search space with less fit solutions, resulting in convergence to global minima in 23th generation (confirmed with grid search). Latter variations of the best fitness value around global minima are related

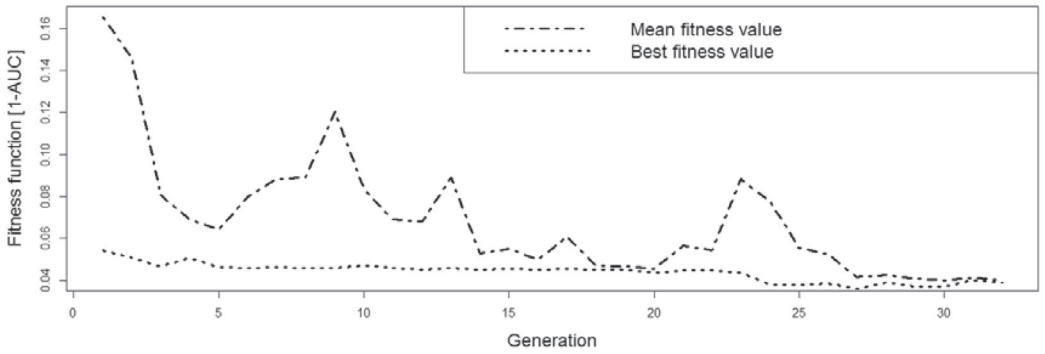

Figure 5. Fitness function development under optimization by GA. Source: Author's own work.

Table 5. Prediction performance metrics and parameter settings for 32nd generation of ANN models.

Number of neurons	Number of epochs	Train Accuracy	CV Accuracy	Test Accuracy	Train AUC	CV AUC	Test AUC	Comp. time [s]
4	350	0.913	0.907	0.904	0.966	**0.961**	**0.957**	2.12
4	350	0.913	0.907	0.904	0.966	0.961	0.957	2.29
4	350	0.913	0.907	0.904	0.966	0.961	0.957	2.22
4	350	0.913	0.907	0.904	0.966	0.961	0.957	2.19
4	300	0.913	0.907	0.905	0.966	0.961	0.957	1.88
4	300	0.913	0.907	0.905	0.966	0.961	0.957	2.01
2	400	0.907	0.905	0.901	0.962	0.961	0.956	1.81
2	350	0.907	0.905	0.901	0.962	0.961	0.956	1.52
2	350	0.907	0.905	0.901	0.962	0.961	0.956	1.61
2	300	0.907	0.905	0.901	0.962	0.961	0.956	**1.48**
2	300	0.907	0.905	0.901	0.962	0.961	0.956	1.57
8	300	0.918	0.904	0.903	0.968	0.959	0.954	3.24
8	300	0.918	0.904	0.903	0.968	0.959	0.954	3.46
128	300	0.917	0.907	0.901	0.964	0.958	0.953	474.35
32	300	0.923	0.908	0.901	0.968	0.956	0.950	23.68
32	400	**0.929**	**0.909**	**0.906**	**0.970**	0.953	0.948	30.30

Source: Author's own work

to changes in random seeds affecting initial set of weights for each ANN.

Prediction performance metrics for all models are calculated as point estimates of mean. Table 5 shows prediction performance metrics and parameter settings of the last generation classifiers, where each row represents ANN model. Training function and regularization parameters are omitted, as they remain same for these models, with training function being Leven-Marquardt (trainlm) and regularization parameter being set to 0. Best results as per column are showed in bold.

As regularization parameter penalizes complexity of ANNs underlying function, it can be concluded that trainlm function, simple ANN classifiers (2–8 neurons in hidden layer) with higher number of training epochs can predict and generalize well in task at hand. This hypothesis is also supported with results of AUC found on test set data. Simple ANNs are preferred as they are less

Table. 6 Prediction performance metrics in terminal nodes of conditional inference tree.

Terminal Node	Train Accuracy	CV Accuracy	Test Accuracy	Train AUC	CV AUC	Test AUC	Comp. time [s]	Number of classifiers
3	0.750	0.753	0.746	0.797	0.801	0.792	**0.83**	12
6	0.848	0.848	0.840	0.911	0.911	0.902	5.97	7
8	0.895	0.895	0.884	0.953	0.953	0.946	5.85	184
9	**0.914**	**0.906**	**0.901**	**0.964**	**0.958**	**0.953**	555.57	141
12	0.773	0.776	0.762	0.890	0.892	0.879	0.93	7
13	0.874	0.874	0.865	0.948	0.948	0.941	3.34	59
14	0.805	0.805	0.795	0.838	0.839	0.832	13.32	8
16	0.637	0.638	0.630	0.683	0.684	0.677	3.62	14
18	0.592	0.593	0.586	0.736	0.738	0.728	1.02	10
20	0.728	0.731	0.719	0.851	0.855	0.840	1.28	11
22	0.816	0.819	0.810	0.914	0.916	0.904	1.53	10
23	0.849	0.850	0.842	0.936	0.937	0.928	4.26	49

Source: Author's own work.

computationally intensive (by an order of magnitude or more in examined generation) and more prone to overfitting, which is crucial property when complexity is not being explicitly penalized in ANN cost function (regularization parameter is set to 0).

For further analysis of prediction performance, conditional inference tree is used to visualize relationship between classifier settings (explanatory variable) and its performance metric (response variable), moreover significance of each split (specific setting) is statistically tested with means of the conditional distribution of linear statistics in permutation test framework (Horton *et al.*, 2006; Horton et al., 2015).

Figure 6 presents classifier results through box-plot charts of 4-fold cross-validation AUC distribution, attached to terminal nodes of conditional inference tree. ANNs settings with trainlm function and regularization parameter 0 results in the best performance while reducing variability of AUC compared to other solutions. This conclusion is strongly supported with low $p<0.001$ (Node 9). Outliers, in the AUC distribution in question, are caused by models with lower number of epochs (numEpochs<300) and lower number of neurons (numNeurons<8).

Other well performing options are based on trainscg function with zero regularization parameter and higher number of epochs (numEpochs>100, Node 8) or settings with trainlm or trainscg function with higher number of neurons ($8<\text{numNeurons}\leq 128$, Node 13). Other considerable possibilities of classification models (Nodes 22, 23) however suffer from higher variability in performance when compared to the prior ones. It can be inducted from presented tree structure that in described experimental context, training function and regularization parameter have significant influence on classifiers AUC performance contrasting other properties of ANN.

Point estimates of mean across classification models in terminal nodes are presented in Table 6. Classifier models in Node 9 outperform every other solution, however they are more computationally intensive by approx. two or more orders of magnitude. This is related to complex ANN models with more neurons in hidden layer and trainlm adaptation function, which scales much worse in terms of computational cost when compared to other training functions. Number of classifiers contained in each terminal node can be related to (i) stability of point

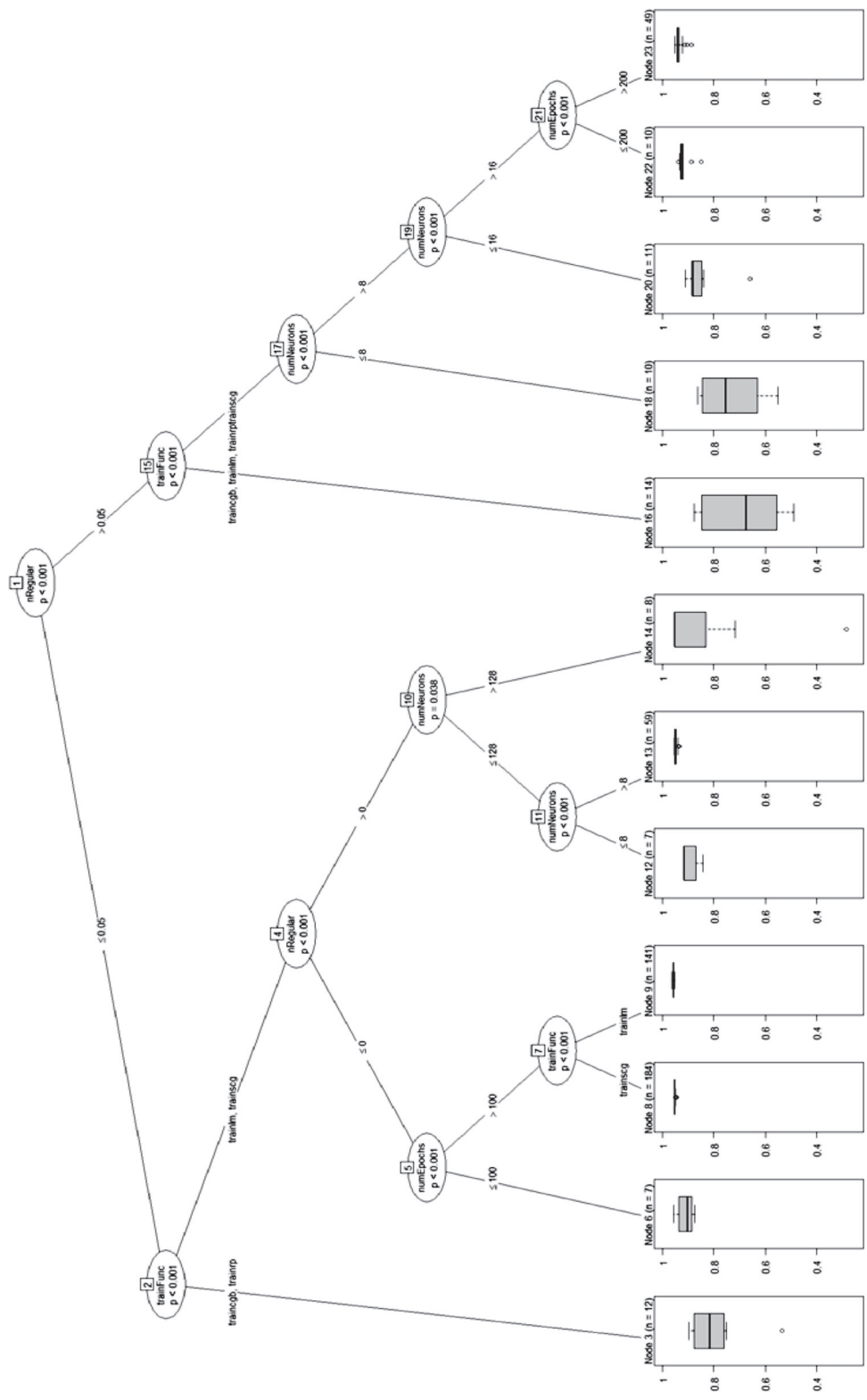

Figure 6. Performance assessed with conditional inference tree structure, distribution AUC mean point estimates, calculated across 4-folds cross-validation, is presented with box-plot charts, R 3.3.0. Source: Author's own work.

estimates and (ii) way how whole parameter space is explored during optimization with regard to parameters considered as important in conditional inference tree structure. It can be inducted that parameter space bounded to branch with high CV AUC is explored more thoroughly with GA, resulting in more models in the branch.

As is indicated in Table 5, listed classifiers perform well and do not undergo overfitting as there are only minor differences among results from training, cross-validation, and testing set. Also, it is shown that there are only minor differences in performance on cross-validation set, favorizing 7-4-2 architecture with 350 training epochs and estimated training and prediction time of approx. 2.2 s. Training function of the model is set to trainlm and regularization parameter to 0.

4. Conclusions and future work

The ability of the company to predict customer churn and retain customers is considered, in saturated markets, as highly valuable competitive advantage. It leads directly to improved cost allocation in customer relationship management activities, retaining revenue and profits in future. It also has several positive indirect impacts such as increasing customer's loyalty, lowering customer's sensitivity to competitors marketing activities, and helps to build positive image through satisfied customers (Colgate, Danaher, 2000; Poel, Lariviere, 2004).

In the paper, machine learning pipeline is proposed to construct and evaluate highly effective and robust classification model, dealing with customer churn prediction, through hyperparameter optimization. Dataset for such task is obtained from e-commerce retail industry. Input data are subject to 2-stage division to firstly ensure there is no data leakage and secondly to properly evaluate performance of the model. Preprocessing of explanatory features is handled through standardization with z-score and PCA. Feedforward 2-layer artificial neural network is used as base classification algorithm, where its architecture and parameter settings are subject to hyperparameter optimization carried out by genetic algorithm, which is adjusted with regard to mixed integer optimization problem. Performance of classification models is assessed with Accuracy and AUC point estimates. Moreover, meta-analysis is originally carried out and visualized with conditional inference tree.

Experimental results show that classification models with trainlm function and regularization parameter set to 0 outperform other models significantly. Results are strongly supported with statistical tests carried out on AUC point estimates from cross-validation set. In addition, point estimates of both Accuracy and AUC on testing set, which is obtained to evaluate performance on real-world dataset with prior selection on cross-validation set, are in line with suggested hypothesis. Considering obtained results and computational intensity, architecture 7-4-2 and 350 training epochs are proposed as suitable for the task at hand.

Several topics can be addressed in further research. New aspects of experimental tuning can be introduced such as multiple objective optimization, explanatory variables selection, transformation process, ensemble learning, and in evaluation of other classification algorithms with accent on interpretability such as logistic regression, decision tree, fuzzy logic *etc.*

As concrete findings are related to e-commerce retail dataset, other domains' datasets might be subject for further exploration and testing. Also, different performance metrics with respect to business context and interpretability might be proposed in future.

References

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer Science Business Media.

Colgate, M. R., Danaher, P. J. (2000). Implementing a Customer Relationship Strategy: The Asymmetric Impact of Poor versus Excellent Execution. *Journal of the Academy of Marketing Science*, 28(3), pp. 375–387. DOI: 10.1177/0092070300283006.

Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2), pp. 311–338. DOI: 10.1016/S0045-7825(99)00389-8.

Deep, K., Singh, K. P., Kansal, M. L., Mohan, C. (2009) A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Applied Mathematics and Computation*, 212(2), pp. 505–518. DOI: 10.1016/j.amc.2009.02.044.

Dostál, P. (2012). *Advanced Decision Making in Business and Public Services*. Brno: Academic Publishing House CERM.

Fausett, L. V. (1994). *Fundamentals of neural networks: architectures, algorithms, and applications*. Upper Saddle River: Prentice Hall.

Fawcett, T. (2006). *An introduction to ROC analysis*. Pattern Recognition Letters, 27(8), pp. 861–874. DOI: 10.1016/j.patrec.2005.10.010.

Garet, J., Witten, D., Hastie, T., Tibshirani, R. (2013). *An introduction to statistical learning: with applications in R*. New York: Springer.

Goldberg, D. E., Deb, K. (1991) A comparison of selection schemes used in genetic algorithms, *Foundations of Genetic Algorithms 1*, FOGA-1, 1, pp. 69–93.

Hadden, J., Tiwari, A., Roy, R., Ruta, D. (2006). Churn Prediction: Does Technology Matter. *International Journal of Intelligent Technology*, (1), pp. 104–110.

Herrera, F., Lozano, M., Verdegay, J. L. (1998). Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. *Artificial Intelligence Review*, 12(4), pp. 265–319. DOI: 10.1023/A:1006504901164.

Hothorn, T., Hornik, K., Zeileis, A. (2006). Unbiased Recursive Partitioning: A Conditional Inference Framework. *Journal of Computational and Graphical Statistics*, 15(3), pp. 651–674. DOI: 10.1198/106186006X133933.

Hothorn, T., Hornik, K., Zeileis, A. (2015). partykit: A Modular Toolkit for Recursive Partytioning in R. *Journal of Machine Learning Research*, 16, pp. 3905–3909. Retrieved from: http://jmlr.org/papers/v16/hothorn15a.html.

Iwata, T., Saito, K., Yamada, T. (2006). Recommendation method for extending subscription periods. In KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, (1), pp. 574–579.

Jain, A. K., Duin, P. W., Jianchang Mao (2000). Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), pp. 4–37. DOI: 10.1109/34.824819.

Kumar, D., Garg, A. (2013). A Study of Data Mining Techniques for Churn Prediction. *International Journal of Science, Engineering and Computer Technology*, 3(1), pp. 1–1. Retrieved from: http://search.proquest.com.ezproxy.lib.vutbr.cz/docview/1515299530/fulltextPDF/99F6B158CE66459BPQ/1?accountid=17115.

Montavon, G., Orr, G. B., Müller, K. -R. (2012). *Neural networks: tricks of the trade*. Berlin: Springer.

Mitchell, M. (1996). *An introduction to genetic algorithms*. Cambridge, Mass.: MIT Press.

Neural Network Toolbox: Computation, Visualization, Programming. (2002). Neural Network Toolbox: Computation, Visualization, Programming. Retrieved from: https://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf.

Ngai, E. W. T., Xiu, L., Chau, D. C. K. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications*, 36(2), pp. 2592–2602. DOI: 10.1016/j.eswa.2008.02.021.

Poel, D., Lariviere, B. (2004). Customer attrition analysis for financial services using proportional hazard models. *European Journal of Operational Research*, 157(1), pp. 196–217. DOI: 10.1016/S0377-2217(03)00069-9.

Rogers, S., Girolami, M. (2012). *A first course in machine learning*. Boca Raton: CRC Press/Taylor.

Skiena, S. S. (1998). *Algorithm design manual*. New York: Springer.

Zou, K. H., O'Malley, A. J., Mauri, L. (2007). Receiver-Operating Characteristic Analysis for Evaluating Diagnostic Tests and Predictive Models. *Circulation*, 115(5), pp. 654–657. DOI: 10.1161/CIRCULATIONAHA.105.594929.

Received: 11. 12. 2016
Reviewed: 22. 5. 2017
Accepted: 24. 5. 2017

Ing. Martin Fridrich, MSc
Brno University of Technology
Faculty of Business and Management
Department of Informatics
Kolejní 4, 612 00 Brno
Czech Republic
Tel.: +420 608 482 996
E-mail: fridrichmartin@yahoo.com